



# Source Analysis with *Fermi* LAT Data

Aous Abdo

< [Aous.Abdo@nrl.navy.mil](mailto:Aous.Abdo@nrl.navy.mil) >

National Academy of Sciences &

Naval Research Laboratory

Washington DC



# Overview

---

- ❖ Detection, flux determination and spectral modeling of Fermi LAT sources is accomplished by a maximum likelihood optimization technique.
- ❖ The likelihood analysis is available in two modes, binned and unbinned. I will be describing the unbinned analysis in this tutorial.
- ❖ Full documentations available at:  
< [http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/likelihood\\_tutorial.html](http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/likelihood_tutorial.html) >



# Unbinned Likelihood

---

## ❖ Assuming that one has:

- Event data file
- Spacecraft data file

## ❖ Steps of the unbinned likelihood analysis:

1. Make the appropriate subselections on the event data file
2. Make counts maps of the region of interest
3. Make an exposure map
4. Obtain the latest isotropic and diffuse models.
5. Create a source model
6. Run the likelihood analysis
7. Make test-statistic maps



# Prepare The Data

---

## 1. Data selection:

- To perform the likelihood analysis a large ROI (10 - 20) degrees should be selected. This is especially important for sources in the Galactic plane.
- We will use `gtselect` to select events in a 10 degree ROI around the CTA1 pulsar:

```
$gtselect infile=@events.txt outfile=FT1.fits ra=1.7565  
dec=73.02552 rad=10.0 tmin= 292937396 tmax=293543047  
emin=100 emax=300000 evclsmin=3 evclsmax=3 zmax=105
```

- list of event files is contained in the text file `events.txt`
- We will perform the likelihood analysis on one week of data.
- Select diffuse Class events



# Prepare The Data

## 2. Run gtmktime:

```
$ gtmktime
Spacecraft data file[]L100422151847E0D2F37E30_SC00.fits
Filter expression[] DATA_QUAL==1 && LAT_CONFIG==1 &&
ABS(ROCK_ANGLE)<52 && IN_SAA!=T
Apply ROI-based zenith angle cut[yes]
Event data file[]FT1.fits
Output event file name[]FT1_gtmktime.fits
```

- ❖ Exclude time periods when some spacecraft event has affected the quality of the data selecting data quality equal 1 (DATA\_QUAL==1).
- ❖ Exclude times when spacecraft was in the SAA
- ❖ Select LAT configuration intended for science analysis (exclude calibration runs and so on.)



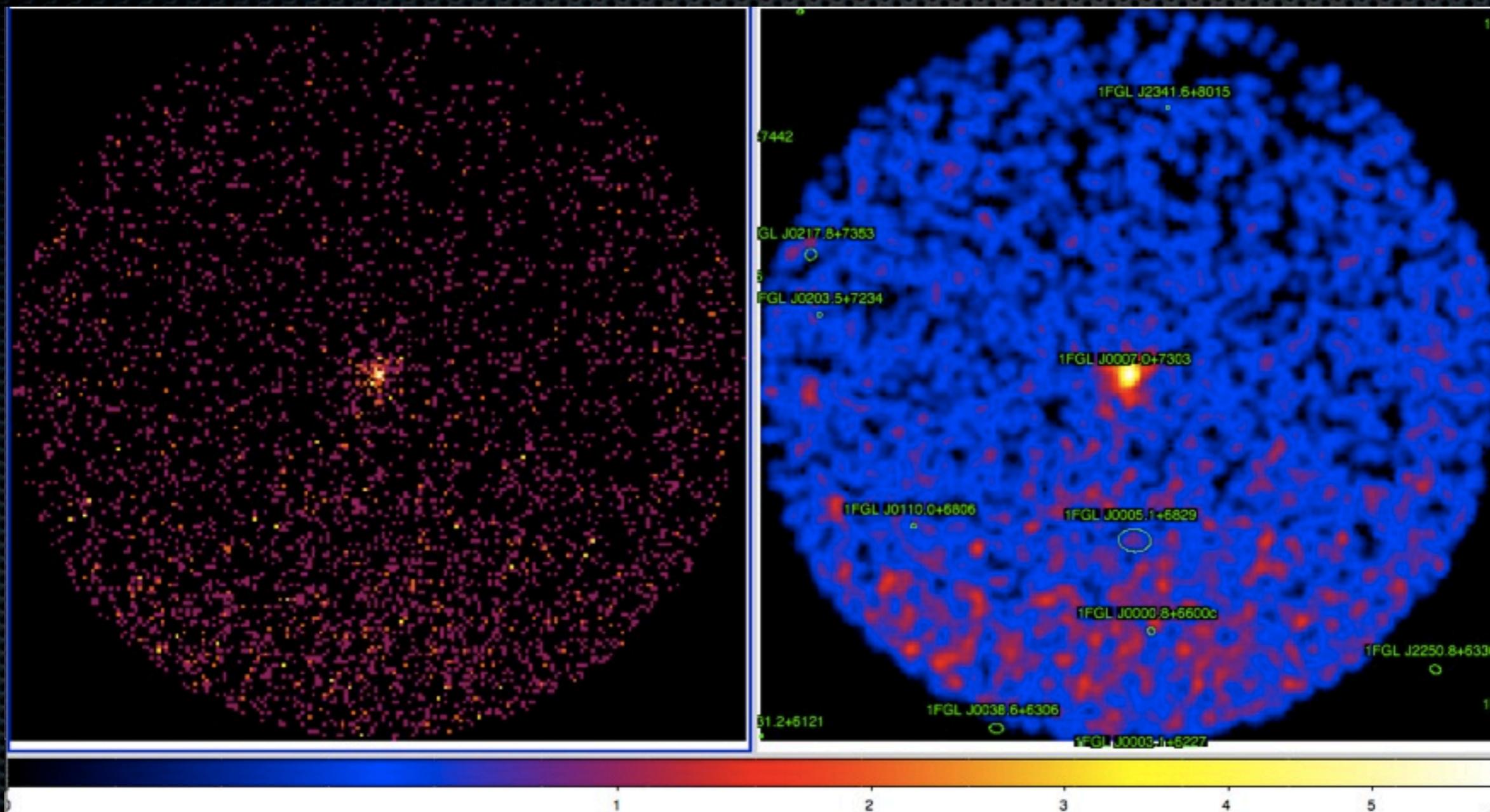
# Make Counts Map

- ❖ Now lets look at the region around our source and see how it looks like for the time period we selected

```
$ gtbin
This is gtbin version v2r2p3
Type of output file (CCUBE|CMAP|LC|PHA1|PHA2) [ ]CMAP
Event data file name[ ]FT1_gtmktime.fits
Output file name[ ]FT1_gtmktime_cmap.fits
Spacecraft data file name[ ]
Size of the X axis in pixels[ ]200
Size of the Y axis in pixels[ ]200
Image scale (in degrees/pixel)[ ]0.1
Coordinate system (CEL - celestial, GAL -galactic) (CEL|GAL) [ ]CEL
First coordinate of image center in degrees (RA or galactic l)[ ]1.7565
Second coordinate of image center in degrees (DEC or galactic b)[ ]73.05225
Rotation angle of image axis, in degrees[ ]0
Projection method e.g. AIT|ARC|CAR|GLS|MER|NCP|SIN|STG|TAN: [ ]AIT
```



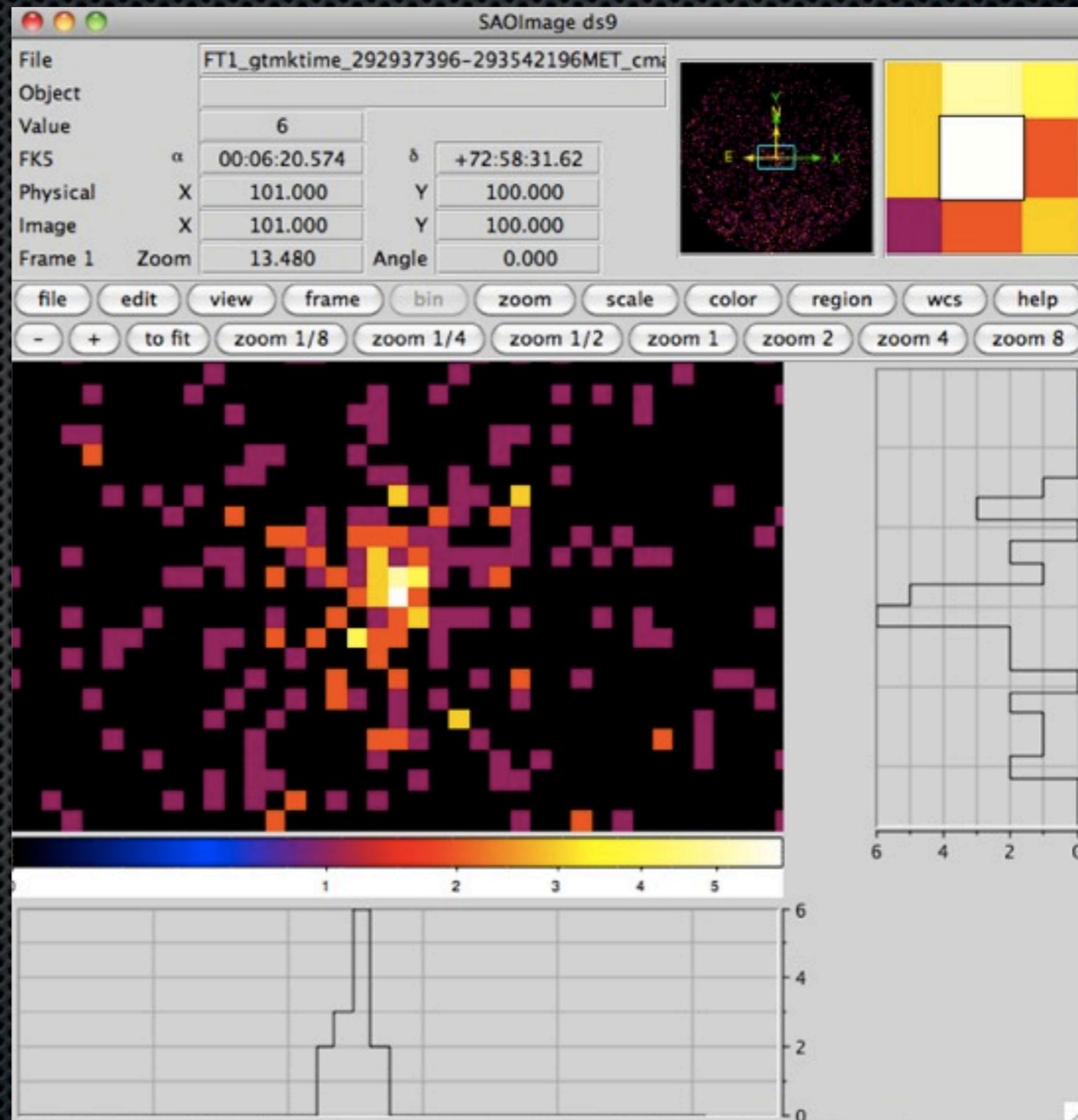
# Counts Map





# Counts Map

- ❖ ~100 photons from this relatively bright gamma-ray pulsar survived our cuts over the period of one week!





# Make An Exposure Map

---

1. First we will generate an exposure cube with gtlcube:

```
$ gtlcube
Event data file[] FT1_gtmktime_292937396-293542196MET.fits
Spacecraft data file[] L100422151847E0D2F37E30_SC00.fits
Output file[expCube.fits] expCube_292937396-293542196MET.fits
Step size in cos(theta) (0.:1.) [0.025]
Pixel size (degrees)[1]
Working on file L100422151847E0D2F37E30_SC00.fits
.....!
```

2. Then we will use the exposure cube we made to make the exposure map with gtexpmap -->



# Make An Exposure Map

- `gtexpmap` creates an exposure map based on the event selection used on the input photon file and the livetime cube.
- The exposure map must be recalculated if the ROI, zenith, energy selection or the time interval selection of the events is changed.

```
$ gtexpmap
The exposure maps generated by this tool are meant
to be used for *unbinned* likelihood analysis only.
Do not use them for binned analyses.
Event data file[] FT1_gtmktime_292937396-293542196MET.fits
Spacecraft data file[] L100422151847E0D2F37E30_SC00.fits
Exposure hypercube file[] expCube_292937396-293542196MET.fits
output file name[] expMap_292937396-293542196MET.fits
Response functions[P6_V3_DIFFUSE]
Radius of the source region (in degrees)[30] 15
Number of longitude points (2:1000) [120] 60
Number of latitude points (2:1000) [120] 60
Number of energies (2:100) [20]
The radius of the source region, 15, should be significantly larger (say by 10
deg) than the ROI radius of 10
Computing the ExposureMap using expCube_292937396-293542196MET.fits
.....!
```

Source region radius  
bigger than acceptance  
cone radius



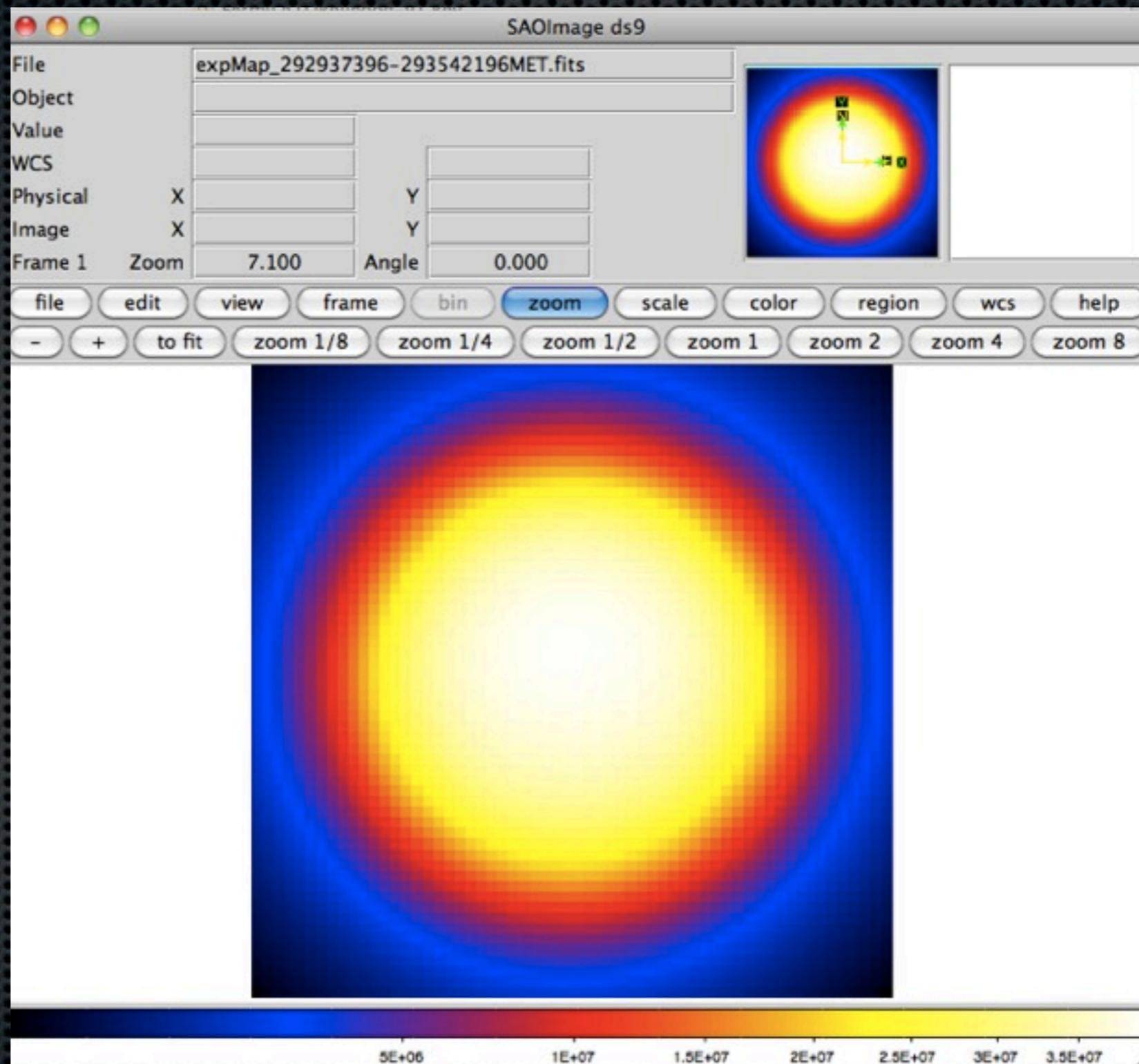
# Make An Exposure Map

---

- Note that we selected for the exposure map a source radius that is larger than the acceptance cone radius.
- This is necessary to ensure that photons from sources outside the ROI are accounted for owing to the size of the instrument point-spread function especially at lower energies
- Half-degree pixels are a nominal choice; smaller pixels should result in a more accurate evaluation of the diffuse source fluxes but will also make the exposure map calculation itself lengthier.
- The number of energies specifies the number of logarithmically spaced intervals bounded by the energy range given in the DSS keywords. A general recommendation is 10 bins per decade. This is sufficient to accommodate the change in effective area with energy near 100 MeV.

# Exposure Map

- ❖ Exposure map of the 15 degree region around CTA1 for the first energy bin (100-152 MeV)
- ❖ Scale is in  $\text{cm}^2 \text{ s}$





# Obtain The Latest Isotropic And Diffuse Models

---



- ❖ To account for all sources in the region of interest we need to model the Galactic diffuse and the isotropic emissions.
- ❖ Those models are available here:  
< <http://fermi.gsfc.nasa.gov/ssc/data/access/lat/BackgroundModels.html> >
- ❖ Use the “gll\_iem\_v02.fit” file to model the Galactic diffuse and the “isotropic\_iem\_v02.txt” file to model the isotropic emission.
- ❖ Details of these models are available on the same link above.



# Create A Source Model

---

- ❖ In addition to the Galactic diffuse and the isotropic models we need to have in our source model all the Fermi LAT point sources.
- ❖ For this we need to download the latest Fermi LAT catalog from:  
[http://fermi.gsfc.nasa.gov/ssc/data/access/lat/1yr\\_catalog/](http://fermi.gsfc.nasa.gov/ssc/data/access/lat/1yr_catalog/)
- ❖ One then needs to extract the point sources in the source region. For this we will use the contributed python script `xmlModelEditor.py` with the catalog in its xml format (file “`gll_psc_v02.xml`”)



# xmlModelEditor.py

---

- ❖ Reads a Fermi xml model file and creates a subset of the given file around a given source with a user-specified ROI. The user can specify an ROI in which some of the spectral parameters of a given source are kept free. The user can also add the standard diffuse models or specify his own.

```
$ xmlModelEditor.py --ra 1.7565 --dec 73.02552 -a 15 -i ./Catalog/  
gll_psc_v02.xml  
You didn't specify a name for the output file. I will be using "srcmodel.xml"  
as the name of the file  
There are 25 sources in the xml file
```

- ❖ This will create an xml file “srcmodel.xml” that contains all 25 point sources in the 1FGL catalog around the CTA1 pulsar in a 15 degree region.



# xmlModelEditor.py

---

- ❖ To add the diffuse models just use the `--diff` option and point to the diffuse model files with the `--gal` and the `--egal` options.
- ❖ We will free the integral and spectral index parameters for all sources that are within 5 degrees of our source. We use the `--roifree` option for this.
- ❖ The selection of 5 degrees in the “free spectral parameter” region is arbitrary but in general one needs not to go below 5 degrees of the source. Freeing all sources in the ROI will make `gtlike` slow.

```
$ xmlModelEditor.py --ra 1.7565 --dec 73.02552 -a 15 --roifree 5 --diff --  
gal ./BackgroundModels/gll_iem_v02.fit --egal ./BackgroundModels/  
isotropic_iem_v02.txt -i ./Catalog/gll_psc_v02.xml  
You didn't specify a name for the output file. I will be using "srcmodel.xml"  
as the name of the file  
There are 25 sources in the xml file
```



# xmlModelEditor.py

```
Terminal Shell Edit View Window Help
abdo@heselin:/Analysis/abdo/tutorial — ssh — 144x42
bash bash labuser@ce...files — bash abdo@hesel...torial — ssh
<?xml version="1.0" ?>
<source_library title="source library">
  <source name="GAL_v02" type="DiffuseSource">
    <spectrum type="ConstantValue">
      <parameter free="1" max="10.0" min="0.0" name="Value" scale="1.0" value="1.0"/>
    </spectrum>
    <spatialModel file="/Projects/GLAST/share/DiffuseModel/gll_iem_v02.fit" type="MapCubeFunction">
      <parameter free="0" max="1000.0" min="0.001" name="Normalization" scale="1.0" value="1.0"/>
    </spatialModel>
  </source>
  <source name="EG_v02" type="DiffuseSource">
    <spectrum file="/Projects/GLAST/share/DiffuseModel/isotropic_iem_v02.txt" type="FileFunction">
      <parameter free="1" max="1000.0" min="1e-05" name="Normalization" scale="1.0" value="1.0"/>
    </spectrum>
    <spatialModel type="ConstantValue">
      <parameter free="0" max="10.0" min="0.0" name="Value" scale="1.0" value="1.0"/>
    </spatialModel>
  </source>
  <source Flux="2.77998983804e-08" FluxError="6.93364027203e-09" Flux_Density="4.70280996126e-13" Npred="399.757568524" Pivot_Energy="2576.31336
508" TS_value="183.835060644" Unc_Flux_Density="5.88526195476e-14" energyFlux="2.19318063339e-05" energyFluxError="2.88332147864e-06" name="1FGL
J0035.9+5951" type="PointSource">
    <spectrum normPar="Integral" type="PowerLaw2">
      <parameter error="0.06933435709" free="0" max="1000" min="1e-05" name="Integral" scale="1e-07" value="0.2773528172"/>
      <parameter error="0.09821376377" free="0" max="0" min="-5" name="Index" scale="1" value="-1.947457177"/>
      <parameter free="0" max="300000" min="20" name="LowerLimit" scale="1" value="100"/>
      <parameter free="0" max="300000" min="20" name="UpperLimit" scale="1" value="100000"/>
    </spectrum>
    <spatialModel type="SkyDirFunction">
      <parameter free="0" max="360" min="-360" name="RA" scale="1" value="8.989"/>
      <parameter free="0" max="90" min="-90" name="DEC" scale="1" value="59.8523"/>
    </spatialModel>
  </source>
  <source Flux="1.06767451985e-07" FluxError="1.1134944387e-08" Flux_Density="1.23892384516e-11" Npred="1214.21195474" Pivot_Energy="749.9146792
25" TS_value="259.699544154" Unc_Flux_Density="9.40153714249e-13" energyFlux="3.66747724772e-05" energyFluxError="2.79087830006e-06" name="1FGL
J0205.6+6449" type="PointSource">
    <spectrum normPar="Integral" type="PowerLaw2">
      <parameter error="0.1113449751" free="0" max="1000" min="1e-05" name="Integral" scale="1e-07" value="1.064237943"/>
      <parameter error="0.05615994244" free="0" max="0" min="-5" name="Index" scale="1" value="-2.364566708"/>
      <parameter free="0" max="300000" min="20" name="LowerLimit" scale="1" value="100"/>
      <parameter free="0" max="300000" min="20" name="UpperLimit" scale="1" value="100000"/>
    </spectrum>
  </source>
</source_library>
--More-- (10%)
```



# Source Model

- ❖ If further editing of the xml source model file is required, one can use the Science Tool modeeditor:

add or remove sources

edit source parameters,  
name, fit parameters,  
scaling ...

for models that require a  
FITS image (extended  
sources, SNR, Galactic  
diffuse)

The screenshot shows the 'python' application window titled 'ModeEditor (/Analysis/abdo/psrsearch/CTA1/CTA1ll/spectrum/srcmodel.xml)'. The window has a menu bar with 'File', 'Edit', 'Source', and 'Help'. The main area is divided into two panes. The left pane shows a list of sources with their names and coordinates. The right pane shows the details for the selected source '1FGL J0035.9+5951'. The 'Source Name' field is filled with '1FGL J0035.9+5951' and the 'Source Type' is 'PointSource'. The 'Spectrum Type' is 'PowerLaw2'. Below this, there are two tables for fit parameters. The first table is for the 'PowerLaw2' model, and the second is for the 'SkyDirFunction' spatial model. Blue arrows point from the text on the left to the corresponding elements in the interface.

name	value	scale	min	max	free
Integral	0.277352817	1e-07	1e-05	1000.0	<input type="checkbox"/>
Index	-1.94745717	1.0	-5.0	0.0	<input type="checkbox"/>
LowerLimit	100.0	1.0	20.0	300000.0	<input type="checkbox"/>
UpperLimit	100000.0	1.0	20.0	300000.0	<input type="checkbox"/>

name	value	scale	min	max	free
RA	8.989	1.0	-360.0	360.0	<input type="checkbox"/>
DEC	59.8523	1.0	-90.0	90.0	<input type="checkbox"/>



# Run gtlike

## ❖ Now we run gtlike:

make a plot of the fitted  
and modeled spectra

name of output file to  
write the fitted model to

```
$ gtlike refit=yes plot=yes sfile=CTA1-out.xml  
Statistic to use (BINNED|UNBINNED) [UNBINNED] \  
Spacecraft file[L100422151847E0D2F37E30_SC00.fits]  
Event file[FT1_gtmktime_292937396-293542196MET.fits]  
Unbinned exposure map[expMap_292937396-293542196MET.fits]  
Exposure hypercube file[expCube_292937396-293542196MET.fits]  
Source model file[CTA1-model.xml]  
Response functions to use[P6_V3_DIFFUSE] Optimizer (DRMNFB|NEWMINUIT|MINUIT|  
DRMNGB|LBFGS) [NEWMINUIT]
```

model xml file we made  
with xmlModelEditor.py

- ❖ gtlike generates two output files: results.dat with the results of the fit, and counts\_spectra.fits with predicted counts for each source for each energy bin.



# gtlike

- 
- 
- 

PSRJ0007+7303:

Prefactor: 10.3479 +/- 1.801

Index1: -1.49866 +/- 0.165733

Scale: 1000

Cutoff: 3.83795 +/- 1.4262

Index2: 1

Npred: 240.637

ROI distance: 0.02673

TS value: 556.776

WARNING: Fit may be bad in range [100, 149.23] (MeV)

WARNING: Fit may be bad in range [495.934, 740.083] (MeV)

WARNING: Fit may be bad in range [1648.13, 2459.51] (MeV)

Total number of observed counts: 4591

Total number of model events: 4591.66

-log(Likelihood): 46460.06264

Writing fitted model to CTA1-out.xml

Observed and modeled counts shouldn't be very different. The similar they are the better the fit we have



# gtlike

## ❖ results.dat file:

parameter has been fitted

```
'PSRJ0007+7303': {'Prefactor': '10.3479 +/- 1.801',  
'Index1': '-1.49866 +/- 0.155733',  
'Scale': '1000',  
'Cutoff': '3.83795 +/- 1.4262',  
'Index2': '1',  
'Npred': '240.637',  
'ROI distance': '0.02673',  
'TS value': '556.776',  
},  
'PSRJ0205+6449': {'Prefactor': '0.931836',  
'Index1': '-2.34344',  
'Scale': '1000',  
'Cutoff': '37.7212',  
'Index2': '1',  
'Npred': '1.66508',  
},
```



# gtlike

- ❖ In the first plot the counts/MeV vs MeV are plotted. The points are the data from the counts\_spectra.fits file, and the lines are the models. Error bars on the points represent  $\sqrt{N_{\text{obs}}}$  in that band, where  $N_{\text{obs}}$  is the observed number of counts. The black line is the sum of the models for all sources. The colored lines follow the sources in your results.dat file as follows:

- Black - summed model
- Red - first source in results.dat file
- Green - second source
- Blue - third source
- Magenta - fourth source
- Cyan - the fifth source

